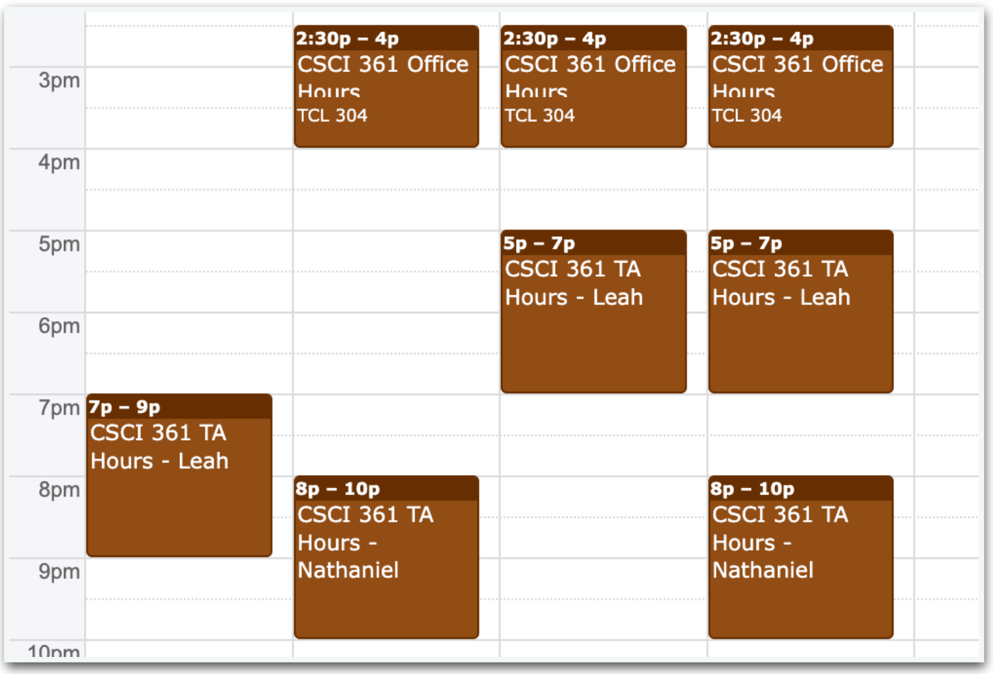
CSCI 361 Lecture 3: Finite Automata

Shikha Singh

Announcements & Logistics

- **HW 1** due Sept 17 (Tuesday)
  - Office hours and TA hours posted on course calendar
- Hand in reading assignment #2
- Pick up reading assignment #3, due at the start of next lecture
- Resources:
  - Lecture 2 handout including proofs on course webpage
  - Board photos posted on GLOW
- New plan for Thurs Sept 19:
  - Class cancelled!
- **Questions?**



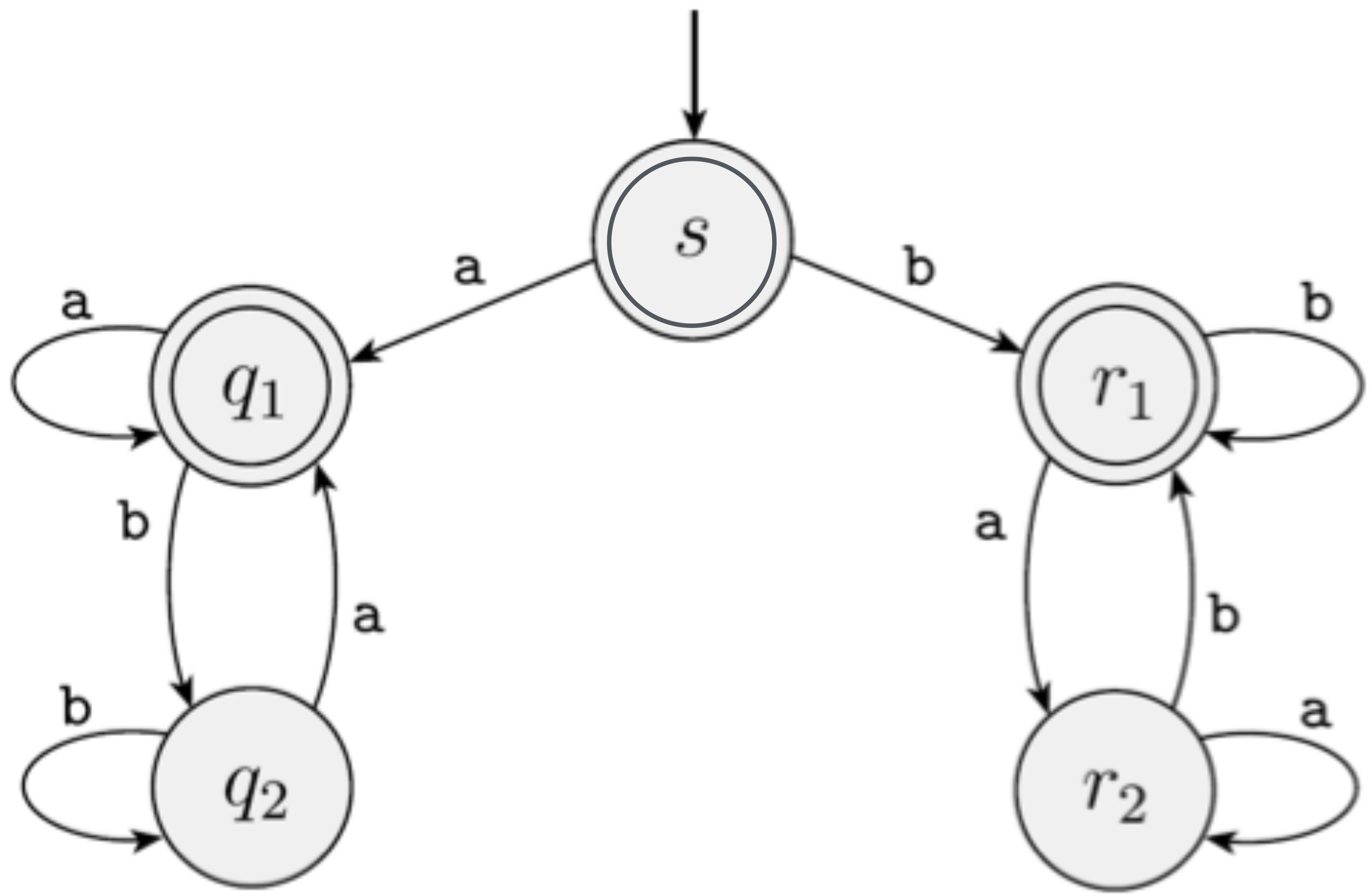
Last Time

- Definitions of finite, countable and uncountable
- Diagonalization argument to prove uncountability
  - Will come back to it when proving undecidability
- Introduced a deterministic finite state automata

Today

- More practice with DFAs and languages recognized by them
- Study regular operators
    - Complement
    - Union/ Intersection
    - Set difference
    - Concatenation
- Introduce a nondeterministic finite automaton: NFA

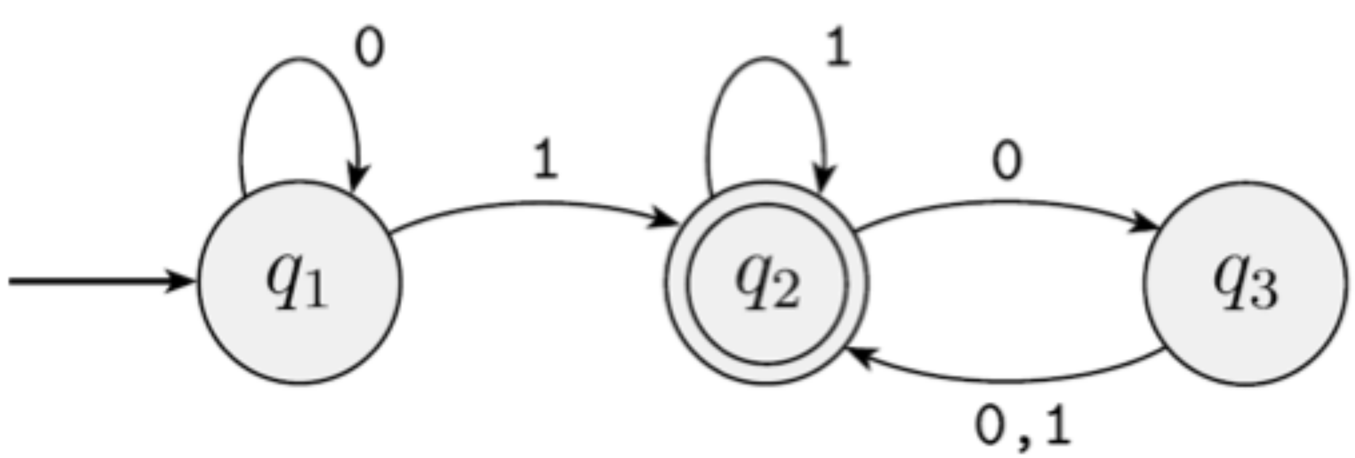
What Language?



Definition of a Finite Automaton

A finite automaton is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where

- Q is a finite set called the states,
- Σ is a finite set called the alphabet,
- $\delta : Q \times \Sigma \to Q$ is the transition function,
- $q_o \in Q$ is the start state and $F \subseteq Q$ is the set of accept states.



Automaton Computation

- Let $M = (Q, \Sigma, \delta, q_0, F)$ be a finite automaton and let $w = w_1 w_2 \cdots w_n$ be a string where each $w_i \in \Sigma$. Then M ***accepts*** w if there is a sequence of $r_0, r_1, \ldots, r_n$ in Q such that
  - $r_0 = q_0$
  - $\delta(r_i, w_{i+1}) = r_{i+1}$ for $i = 0, 1, \ldots, n-1$ and
  - $r_n \in F$

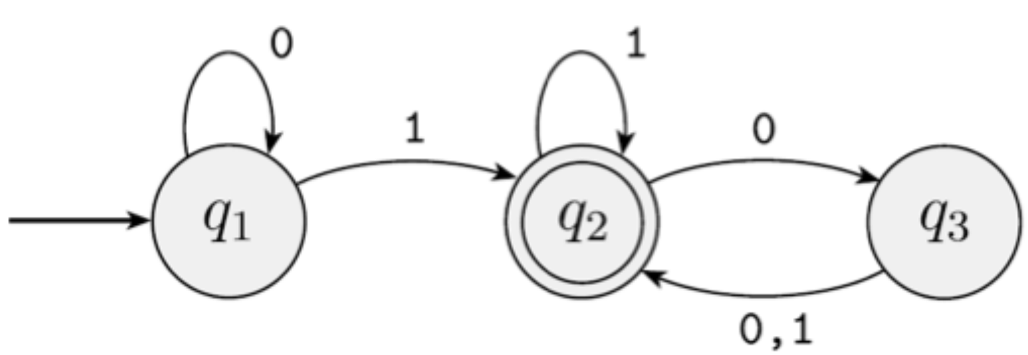
Extended Transition Function

- Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA
- Transition function $\delta : Q \times \Sigma \to Q$ is often extended to $\delta^* : Q \times \Sigma^* \to Q$ where $\delta^*(q, w)$ is defined as the state the DFA ends up in if it starts at q and reads the string w
- Alternate definition of M accepts $w \iff \delta^*(q_0, w) \in F$

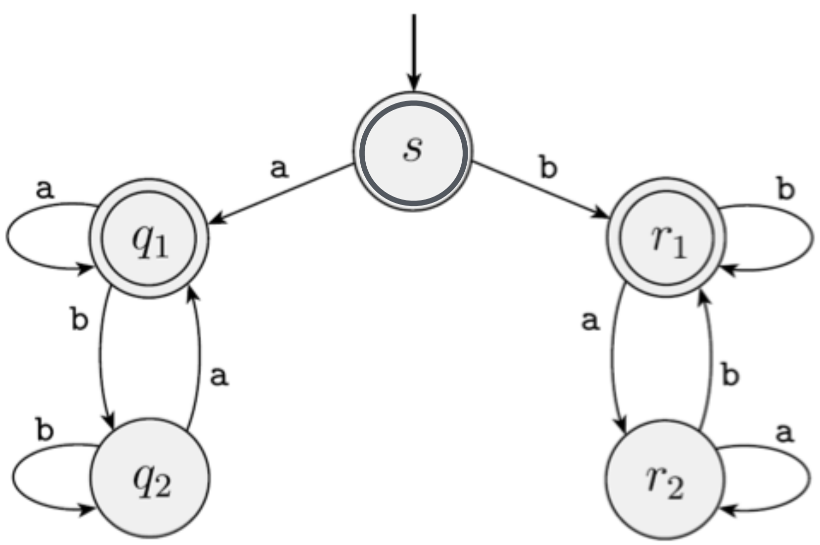
Language of a Machine

- The set of all strings accepted by a finite automaton M is called the language of machine M, and is written $L(M)$.
    - Say M **recognizes** language $L(M)$

M_1



M_2



$L(M_1) = \{w \mid w$ contains at least one 1 and an even number of zeroes follow the last $1\}$

$L(M_2) = \{w \mid w \in \{a, b\}^*$ that starts and ends with the same symbol$\}$

Regular Languages

- **Definition**. A language is called a **regular** language if some deterministic finite automaton recognizes it.

- Thus, to show a language L is regular, we must design a DFA M that recognizes it, that is, $L(M) = L$

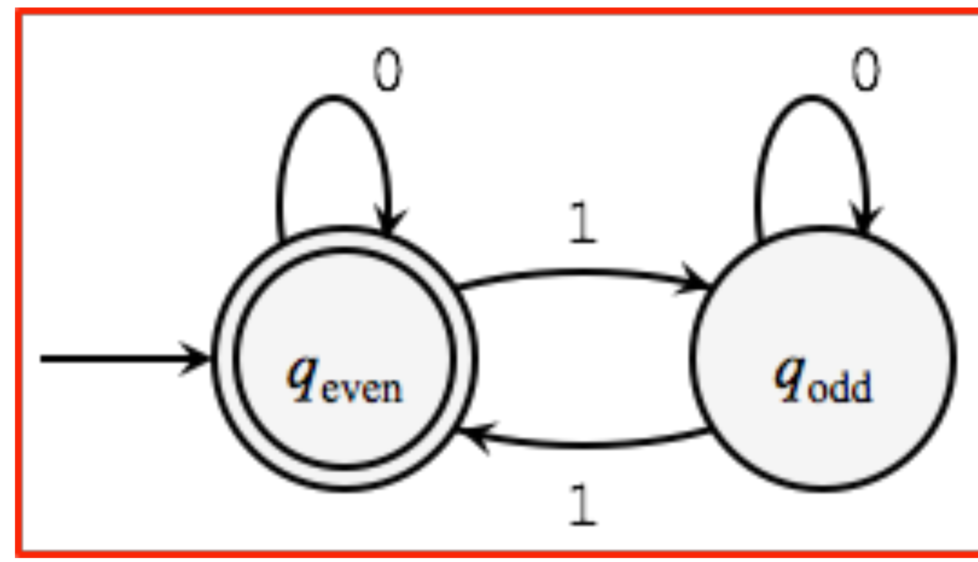
  - M accepts $w \iff w \in L$

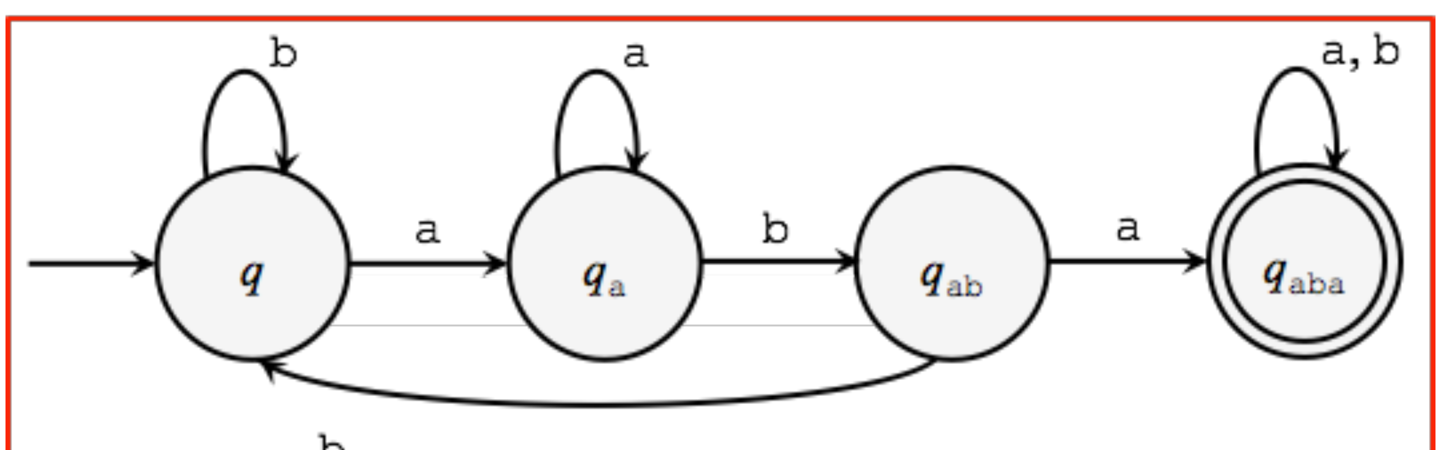
Practice with DFAs

- Show that the following languages are regular by drawing the state diagram of a DFA that recognizes it:
- $\{w \in \{0,1\}^* \mid w$ contains an even number of 1s $\}$
- $\{w \in \{a, b\}^* \mid w$ contains the substring aba $\}$

Class Exercises

- Show that the following are regular:
- $L_1 = \{w \mid w$ is a string of 0s and 1s containing an even number of 1s $\}$
- $L_2 = \{w \mid w$ is a string of as and bs containing the substring aba $\}$





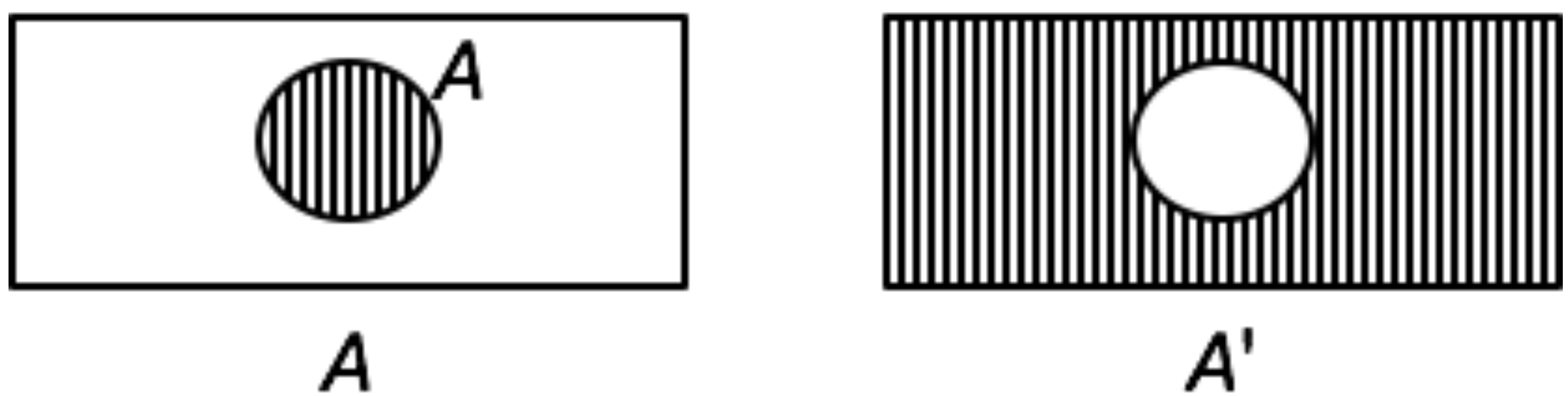
How About These Languages?

- Any similarities?
  - $L_3 = \{w \in \{0,1\}^* \mid w$ contains an **odd number** of 1s $\}$
  - $L_4 = \{w \in \{a,b\}^* \mid w$ **does not contain** the substring aba $\}$

Regular Operations

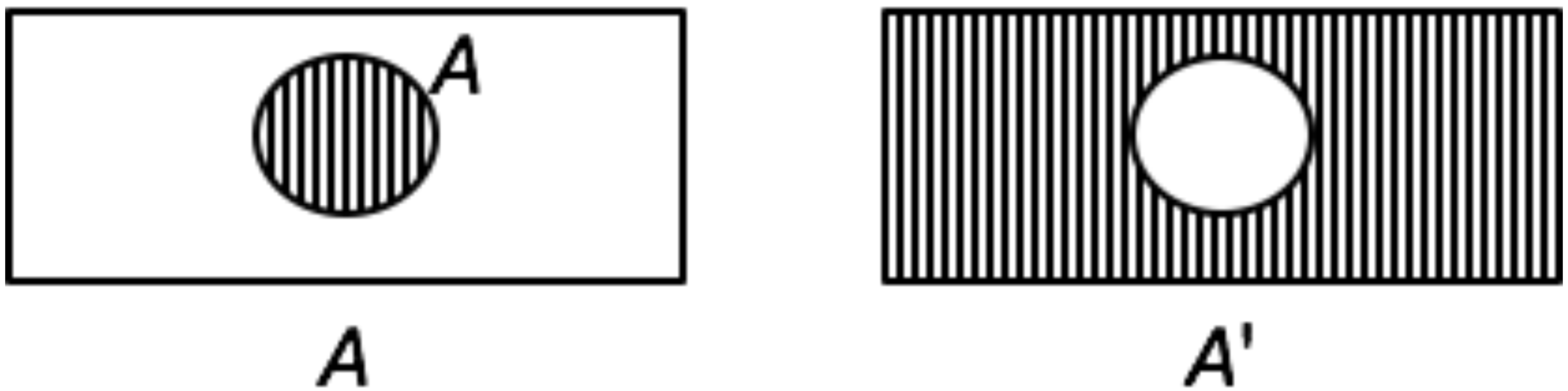
Building New Languages From Old

- Let A be a language on Σ
- Complement of A, denoted $\overline{A} = \{w \in \Sigma^* \mid w \notin A\}$



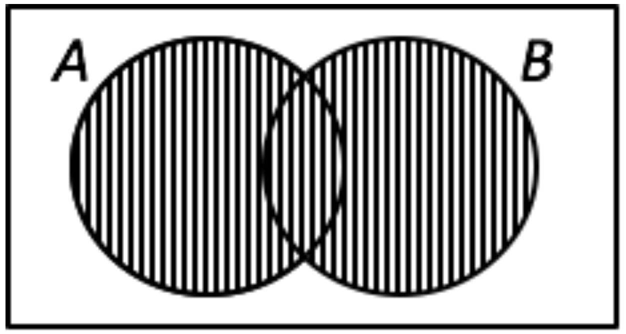
Closed Under Complement

- **Theorem.** The class of regular languages is closed under the complement operation.

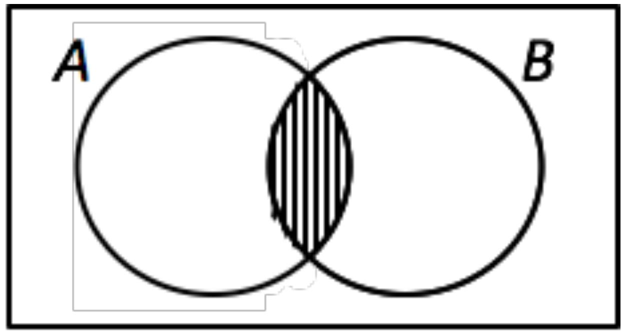


Union and Intersection

- Let A and B be regular languages over Σ.
- Is $A \cup B$ regular?   Is $A \cap B$ regular?



$A \cup B$



$A \cap B$

Closed Under Intersection

Theorem. The class of regular languages is closed under the intersection operation.

Closed Under Union

Theorem. The class of regular languages is closed under the union operation.

Concatenation

- Let A and B be languages over Σ.
- **Definition.** Concatenation of A and B, denoted $A \circ B$ is defined as

$$A \circ B = \{xy \mid x \in A \text{ and } y \in B\}$$

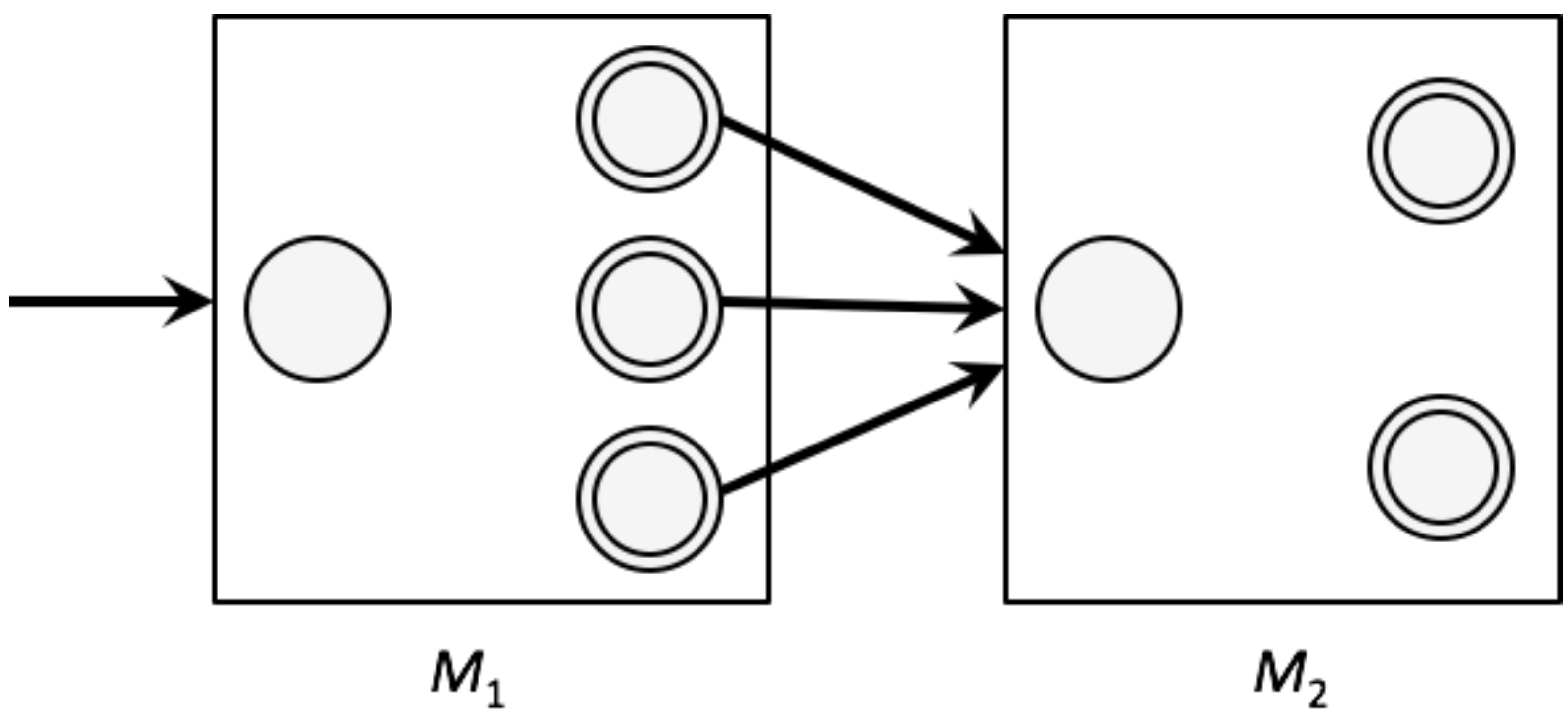
- **Question**. Are regular languages closed under concatenation?

Intuition: Closed Under Concatenation

- Let A and B be languages over Σ.
- **Definition.** Concatenation of A and B, denoted $A \circ B$ is defined as

$$A \circ B = \{xy \mid x \in A \text{ and } y \in B\}$$

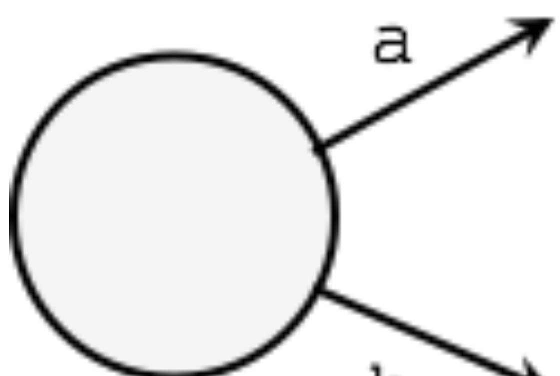
- **Question**. Are regular languages closed under concatenation?



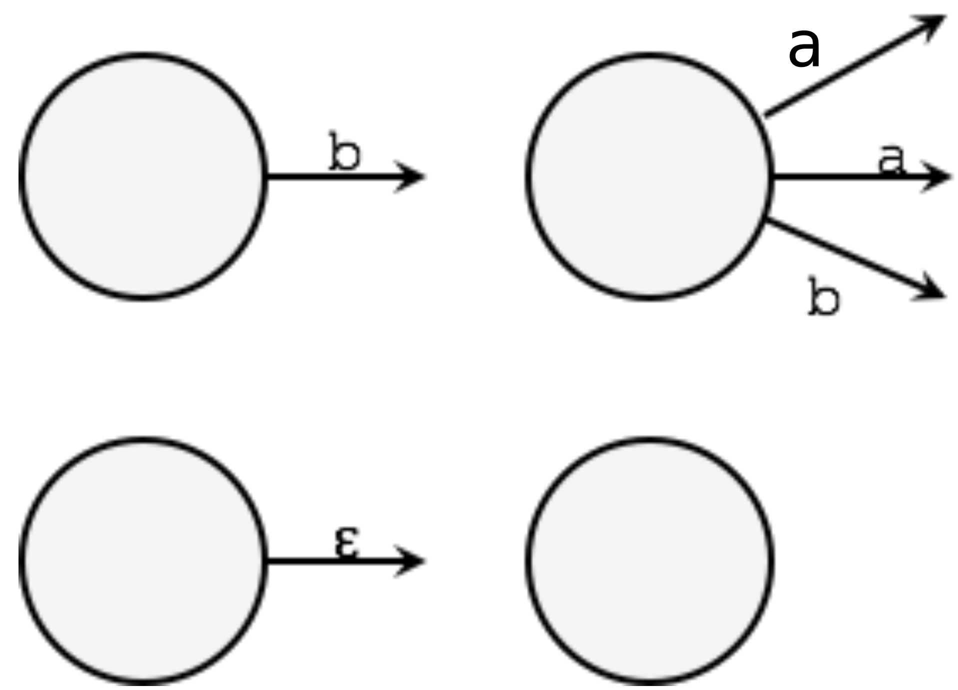
Non-deterministic Finite Automaton (NFA)

Relaxing the Rules

- Deterministic Finite Automaton (**DFA**)

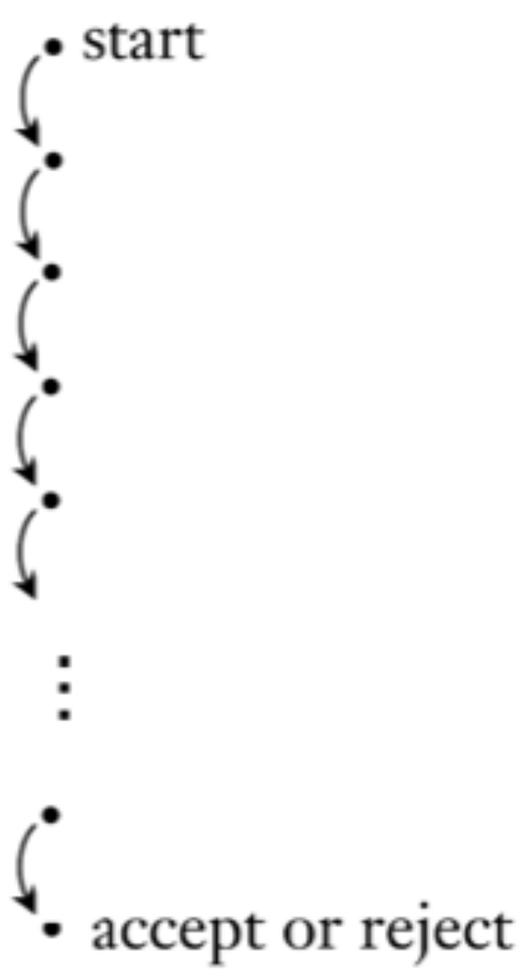


- Non-deterministic Finite Automaton (**NFA**)

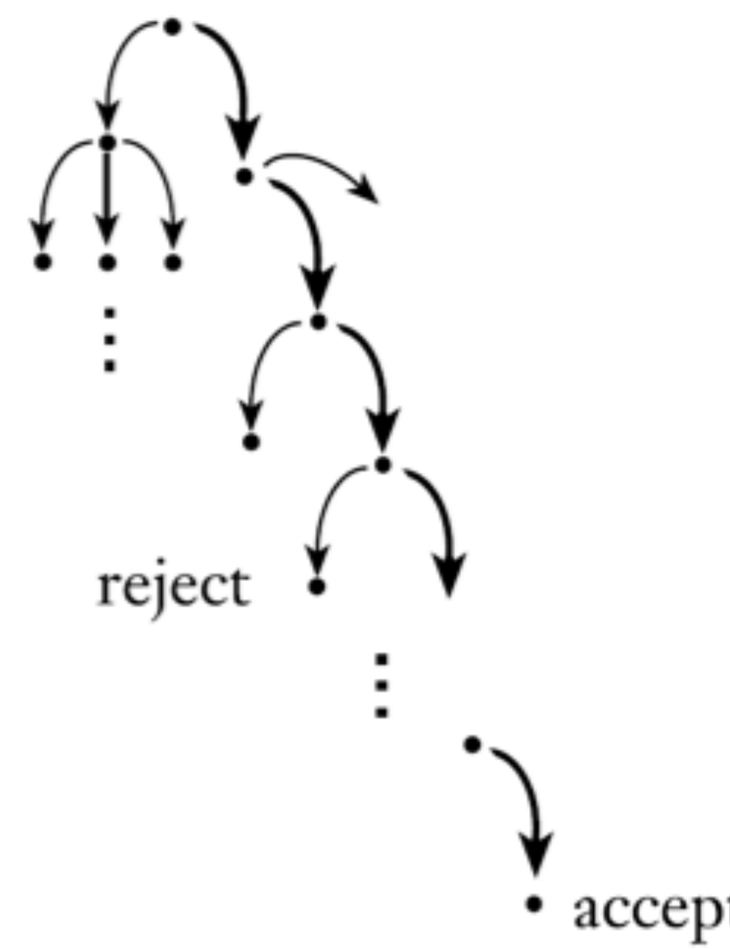


How Does Computation Proceed?

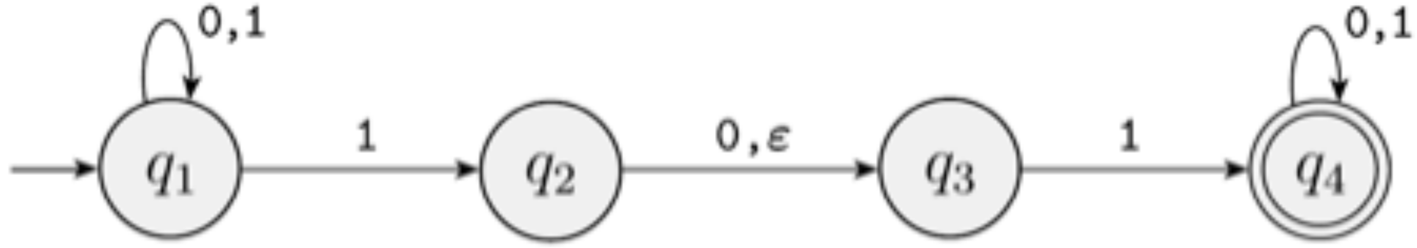
- Deterministic Finite Automaton (**DFA**)



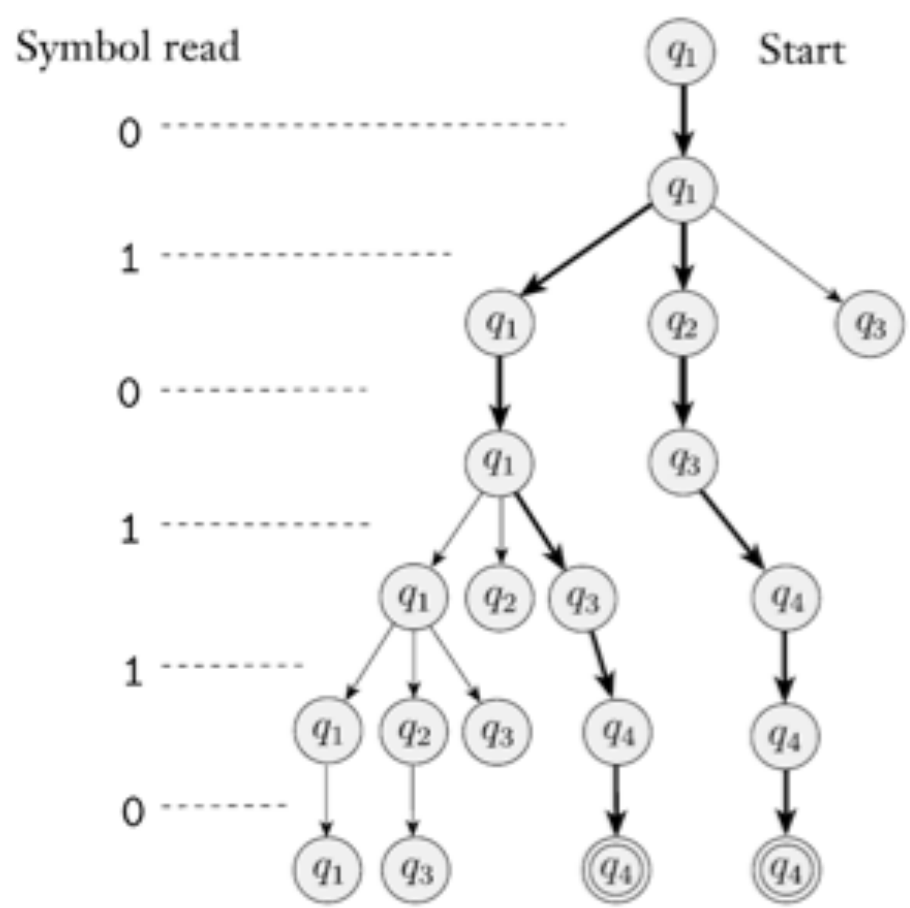
- Non-deterministic Finite Automaton (**NFA**)



Example of NFA N_1 from Sipser



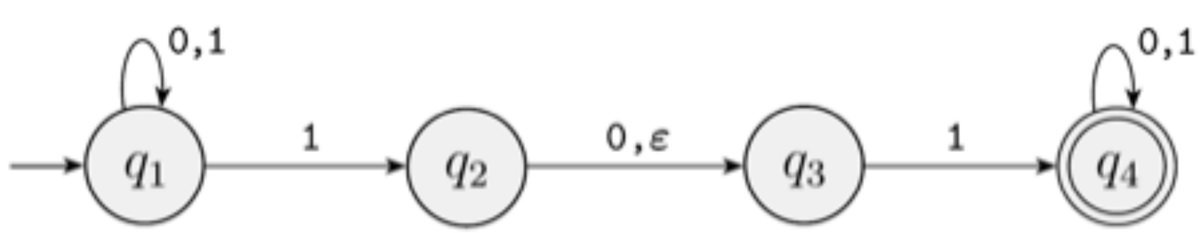
Input: 010110



Formal Definition: NFA

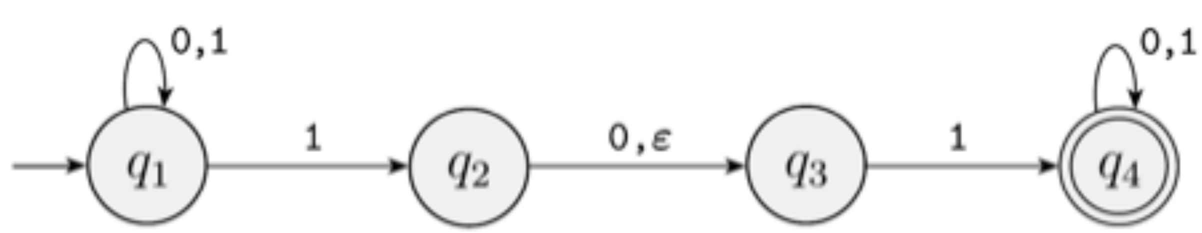
A non-deterministic finite automaton is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where

- Q is a finite set called the **states**,
- Σ is a finite set called the **alphabet**,
- $\delta : Q \times \Sigma_\varepsilon \rightarrow \mathscr{P}(Q)$ is the transition function,
- $q_o \in Q$ is the **start** state and $F \subseteq Q$ is the set of **accept** states.



NFA Computation

- Let $N = (Q, \Sigma, \delta, q_0, F)$ be a non-deterministic finite automaton and let $w = w_1 w_2 \cdots w_n$ be a string where each $w_i \in \Sigma$. Then N ***accepts*** w if there is a sequence of $r_0, r_1, \ldots, r_n$ in Q such that
    - $r_0 = q_0$
    - $r_{i+1} \in \delta(r_i, w_{i+1})$ for $i = 0,1,\ldots,n-1$ and
    - $r_n \in F$



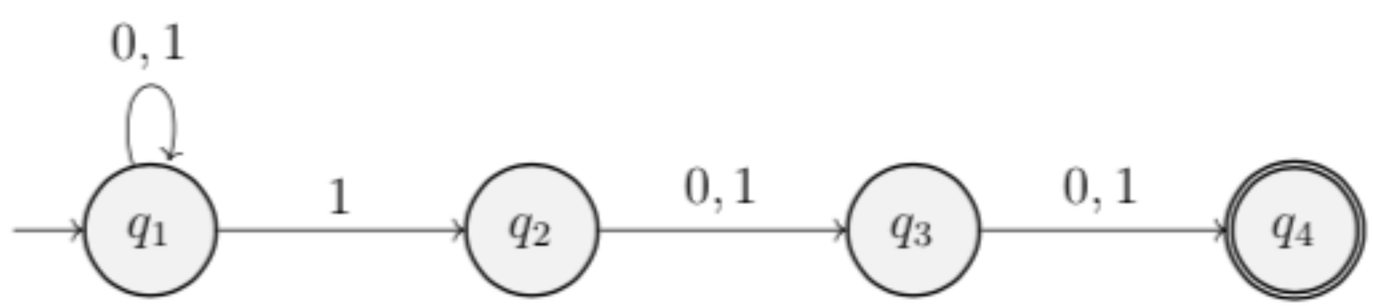
Nondeterminism is Your Friend

- Build an NFA to recognize the following language:
- $L = \{w \mid w \in \{0,1\}^*$ and has a 1 in the 3rd position from the end$\}$

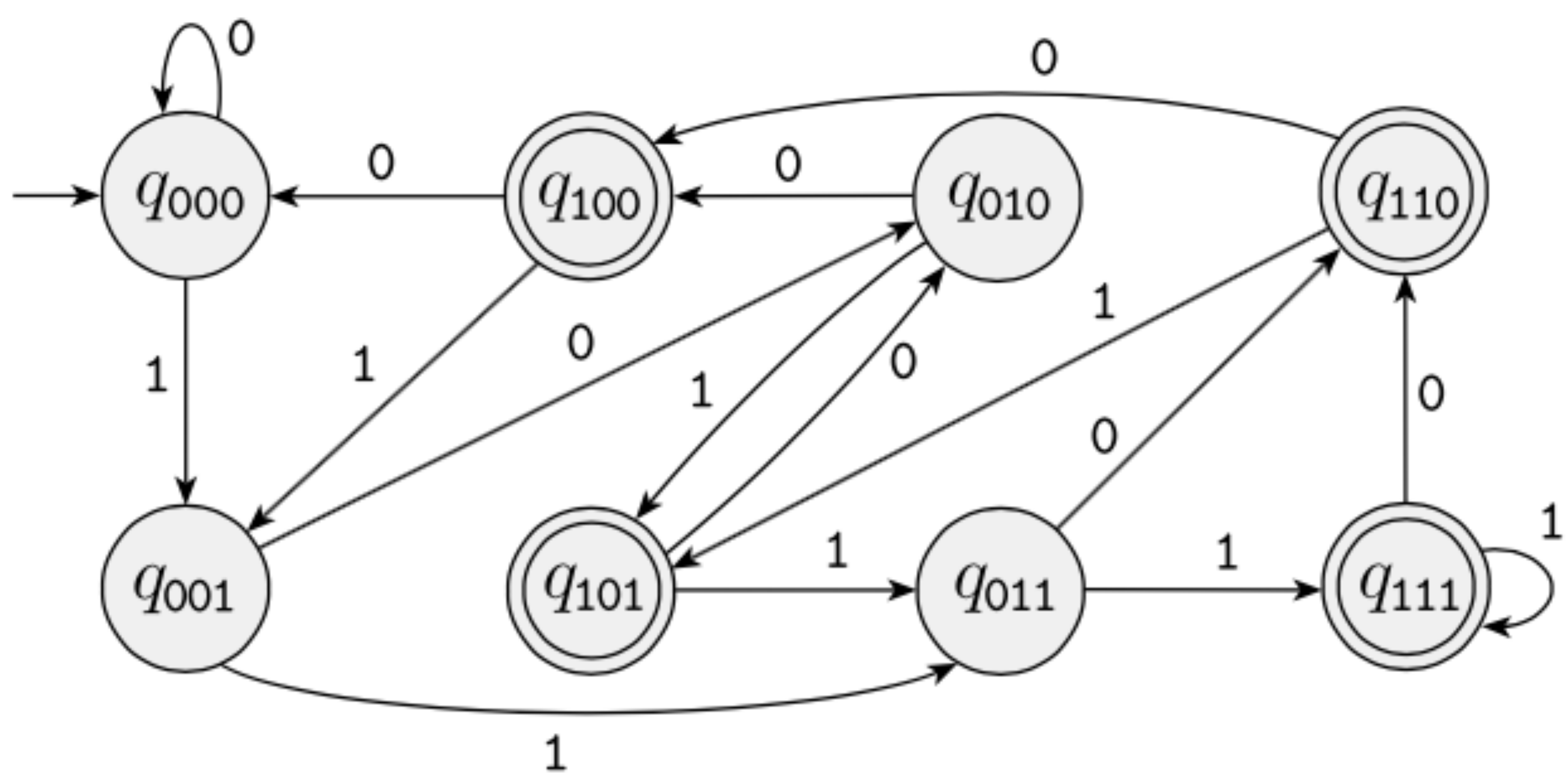
Nondeterminism is Your Friend

- Build an NFA to recognize the following language:
- $L = \{w \mid w \in \{0{,}1\}^*$ and has a 1 in the 3rd position from the end$\}$

NFA



DFA



Kleene Star

- Let A be a language on Σ
- Definition. Kleene star of A, denoted $A*$ is defined as:

$$A* = \{w_1 w_2 \cdots w_k \mid k \geq 0 \text{ and each } w_i \in A\}$$

- **Example**. Suppose $L_1 = \{01, 11\}$, what is $L*$?

- **Question**. Are regular languages closed under Kleene star?

Not All Languages are Regular

- Intuition about regular languages:
  - DFA only has finitely many states, say k
  - Any string with at least k symbols: some DFA state is visited more than once
    - DFA "loops" on long enough strings
  - Can only recognize languages with such nice "regular" structure
- Will see general techniques for showing that a language is not regular
- Classic example of a language that is not regular:
  - $\{w = 0^n 1^n \,|\, n \geq 0\}$ (equal number of 0s and 1s)